Advanced Engineering Technology and Application
An International Journal

http://dx.doi.org/10.12785/aeta/040205

Automation of HCI Engineering processes: System Architecture and Knowledge Representation

M. N. Sheriyev and L. B. Atymtayeva*

Kazakh British Technical University, Almaty, Kazakhstan

Received: 3 Mar. 2015, Revised: 26 Apr. 2015, Accepted: 28 Apr. 2015
Published online: 1 May 2015

Abstract: The aim of this paper is to introduce the methods, techniques and theories concerning web usability and the automation of human-computer interaction (HCI) processes for the intelligent development of web-applications based on the main principles and rules of HCI Engineering. Starting from the description of the traditional ergonomic methods and models we come to the suggestions of the creating the architecture and knowledge base ontology of HCI-Engineering Intelligent Systems that may help to create good and usable web-interfaces.

Keywords: human computer interaction, engineering knowledge, base knowledge, frame model ontology, Intelligent System.

1 Introduction

The user interface is the important part of the direction of Human-Computer Interaction (HCI), which started to be regularly studied since 1960s. According to the recommendations of experts at least 10% of total budget of software development project should be allocated to development of the usable interfaces with enough level of usability. According to statistics at the USA [1].the average improvement of key business indicators belongs to the market of web applications which occupies about 83% of the software development market. This fact gives us evidence of significant economic efficiency of interactive software. However, not all projects related to software development are implemented by using the practical and effective interaction design methods. As a result, the value of the benchmark as the percentage of successful completed tasks for web applications is no more than 81% [1] and for certain categories of users it is 1.5-2 times lower.

We can identify the problem of poor process of the use the practical HCI knowledge which accompanies by significant time costs in searching, interpreting and applying the relevant recommendations or ready design patterns (standard solutions used in the design interfaces) for developers [2]. To solve this problem it was proposed some solutions by development of the different intelligent systems (IS) in order to support the design of interfaces by dividing into elements and organization of recommendations (for example, MetroWeb, BORE systems and etc.) [2]. These systems mostly give the opportunity to automate code generation and the process of the interface validation. The effective combination of these proposed approaches would reduce the time required for searching of appropriate recommendations in design, in finding the usability problems, and in suggesting the associated with them solutions for improvement of the user interfaces.

The aim of this paper is to propose possible solutions in automation of HCI Engineering processes by developing HCI analysis system with elements of intelligent tools for providing the improved human-computer interfaces used in web applications. The knowledge base [22] is the main part of any expert system. In this paper we are going to present the ontology and main architecture of the automated intelligent system that covers as fully as possible the stages of web-applications development process and takes into account the specificity of the HCI interaction in terms of practical knowledge in relevant context.

To achieve **these goals** we follow the next steps:

1. Analyze the structure of knowledge in the HCI area and design of interaction process;
2. The selection of adequate models and tools for knowledge representation;

* Corresponding author e-mail: madi.sheri@gmail.com

3. Development and experimental study of interaction patterns in human-machine interface to identify the profiles of users which have relations to the various aspects of interaction.
4. Building the knowledge base and system architecture for processing the HCI engineering tools in web applications including mechanisms of knowledge storing and assessment of the relevant effectiveness and quality of results.

2 Actual problems of HCI engineering

Human-computer interaction (HCI) is relatively young and a wide scientific section in IT and currently has no finally settled definition. As a working definition we are able to say following: Human Computer Interaction is a discipline that studies the design, evaluation and implementation of interactive computer systems intended for human use, as well as related aspects.

During the recent period starting from 2000 the total number of Internet users in the world increased in more than 5.5 times (more than 2 billion people) [3] and in Kazakhstan the average annual increase achieved 28.8%, and the number of users exceeded 8 million (43% of the adult population) [4]. A similar rapid growth dynamics is also shown on the number of websites: if the total number of web-sites in the world was about 1 million in 1997, then 100 million in 2006 [5] nowadays we have more than 150 million web-sites in Internet world [6].

The quality of interaction is very important for web applications. The famous expert in HCI area J. Nielsen gives the following expression describing the business effect (B) of the e-commerce for web-site [1]

$$B \sim V * C * L, \tag{1}$$

where B is amount of business done by the web- site, V - unique visitors coming to the site, C - conversion rate (the percentage of visitors who become customers); here we can note that the concept of conversion applies not only e-commerce web-sites, but also any web-applications which are used by the different users and L - loyalty rate (the number of the customers that are returned for conducting repeat business) [1].

The quality of usability is very crucial for web-application in e-commerce and e-business and any other e-area. But we can notice that the level of usability in web applications is still relatively low for many systems. There is a tendency to improve this situation. The solutions for practical problems in projects based on the development of web-applications contain the informal and empirical methods, and generally apply the modern approaches in HCI-engineering. We can distinguish the following factors which have impact to the usability improvement methods [18]: task analysis, iterative design and testing with real users.

The aim of task analysis is the identifying the challenges that the user faces and finding the existing or possible solutions. Application of this method is advisable for later formulation of functional requirements and use cases. Iterative design implies a gradual improvement of interface quality. By using this approach the usability problems and poor design solutions may be discovered at early stage of development and can help in minimizing the costs [18]. The most popular method for identifying the problems of human - computer interaction is usability testing.

Usability testing is a set of experimental techniques and methodologies for assessing the degree of usability of a product (web application programming interface, document, etc.) by testing them with real or potential users. One of the benefits of usability testing is its involving of a relatively small number of users. Generalizing data from a large number of projects in IT sector we can conclude that testing with just five users is able to identify more than 80% of the usability problems [7].

3 HCI engineering and software development process

On requirements analysis phase HCI engineering methods aimed at collecting requirements related to interaction with the system and fixing quality targets interface [21].

While making decisions during the project creation interface, developer can based on its own experience and principles described in literature of interaction and practical recommendations, according to context of the project. The search for an existing HCI knowledge in target context of development may creates a substantial labor costs, its correct interpretation and adaptation require a high level of skills of a designer [19]. Software implementation usually misses the area of interest in HCI, because the quality of interface is translated into a rectilinear object code.

Testing the interface can serve two purposes: to identify problem arising from interaction or assessment of its quality performance; to compare different projects interface with each other. Some of testing methods for users can be fully or partially automated. That is embodied in numerous instruments that exist in this area.

Implementation is a set of measures with established software system and support (maintenance). It works on its improvement and development, adapting new conditions and bug fixes. Unlike the previous stages of the cycle of software development, this activity may not be scheduled finite duration and it may continue until users operate the system. At this stage consumes about 60% of the total cost of software development. A main activity is not associated with error correction, as is sometimes supposed with the improvement and future development [9].

After software implementation, developers finally get potentially unlimited opportunities to analyze the behavior of users during the interaction, clarify their needs and iterative improvement of interface. The degree of success of the projected interface (quality) understood as a set of values of some characteristics that are working with the user software. In the next section of our work is a brief overview of user interfaces creating approaches and discusses the concept of quality of interaction.

4 User Interface

Interface looks like a system designed for human-computer interaction including a physical (hardware) and logical (software) components.

One of the main aims of HCI both scientific and practical section is to improve the quality of user interfaces, which is important part of quality software product. Standards, which describe the quality software tools are usually not distinguish the characteristics associated with quality of user interface in a separate category. Standard ISO / IEC 9126-93 [10] identifies the category of "Usability" as "a set of attributes related to the volume of work required for ... express or implied terms of users" without mentioning interface explicitly or interaction. Part 4 of the international standard ISO / IEC TR 9126, dedicated to Quality of Use, gives the following set of characteristics: Effectiveness, Productivity, Safety and Satisfaction [17]. J. Nielsen gives the following five qualitative characteristics of "usability" interface [18]:

1. Learnability - how easily user can start working with the system, seeing it at first time;
2. Efficiency - how to work can be productive when user has mastered it;
3. Memorability - as far as user can easily go back to the effective operation of the system after a long break;
4. Errors - the error rate when operating the system, their severity and ease their correction;
5. Satisfaction - how nice to use the system.

Measurement of learnability and memorability of system in practice is rarely done, because it is the most difficult. Besides these characteristics are relevant not for all software products. Most web applications will not be specifically examined by user and may never be used it again. That's why vast plurality of quantitative degree of usability interface, it is recommended to use the following [11]:

1. Success rate - the percentage of successful completion of user tasks;
2. Performance time - time spent on tasks;
3. Error rate - the level of errors made in process of implementation
4. Subjective satisfaction - subjective user satisfaction.

These set of indicators is widespread for describing quality of user interface (Standard ISO / IEC 25062:2006 [12]) and our work is based on these standards. Despite of relative availability and effectiveness of these methods, the provision of quality of communication is not a trivial task even if we have a team of developers skilled in this area. They can spend many times to perform their functions, especially if group target for the developed software system is special category of users and if these group users are not available for the development team (which may occur during web applications development), etc. Thus, it is important to use software tools that can provide support HCI engineering at various stages of the software development process.

5 IS architecture: input and output information

The intelligent system is designed to support HCI engineering, especially in early stages web applications development. That's why system based on the knowledge of different areas such as HCI interface design, graphic design, usability design, etc. A final decision regarding the interface is accepted by expert, who uses the system. The main component of the output of IS is a set of recommendations in natural language.

As a solution to problem of processing requirements for web application IS offers an analysis of the text itself. Thats why we created Knowledge Base which contains vocabulary terms of subject area in system used for text analysis, indexing recommendations and describe context design.

The existence of KB in intelligence system with controlled vocabulary can describe complex objects of subject area by comparing a list with basic terms. For example, knowledge base with the following recommendation: "Site logo should have a link to home page," knowledge engineer should match terms Logo, Home, Hyperlink". The context of project is described in similar way, but the set of terms formed by an intelligent system automatically based on target user and requirements defined as input information. Comparing these two sets the system is able to determine the relevance of each recommendations. Each recommendation is evaluated as a potential component of "solving the problem" or output list of recommendations.

Validation mechanism and learning for intelligent system is special web portal of knowledge which offers to user ability to assess the applicability of the knowledge contained in IS. Recommendations from HCI area, originally presented in natural language, can be transformed into knowledge, presented in formal rules of inference production model. The architecture of IS for HCI engineering can be represented as shown in **Fig. 1**.

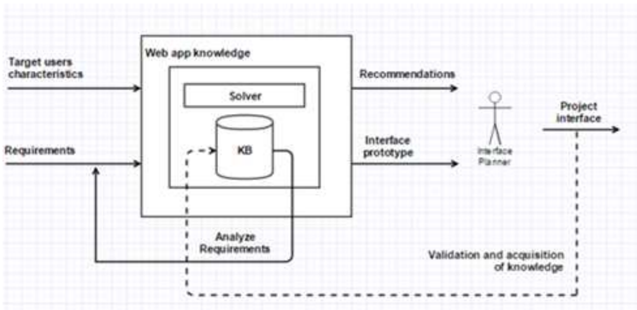

Fig. 1: IS architecture for HCI design support in web applications

6 Knowledge representation models and hybrid model for IS

The core of IS are knowledge base and solver that allows: to compare input information and the information provided in the knowledge base; to receive output information. For knowledge representation we proposed hybrid model, including frame-based ontology which uses object-oriented approach and production model, which allows form implication knowledge.

There are various models of knowledge representation: formal logic, production, framing and semantic network. One of the modern knowledge representations are ontologies, formal explicit description of the terms and relationships between them (defined according to T. Gruber, whose work is significant [20] in using of ontologies in artificial intelligence). To describe the ontology can be used syntax of predicate logic (subset). Under the definition of ontology there are many models of knowledge representation: frames, semantic networks, conceptual maps, etc. In our work we use frame onthology.

Frame [13] is the structure of knowledge, modeling human thinking and corresponding abstract image of any object, phenomenon, events or process. A main feature of frame model is that initial slots can be filled with ”absence of tasks”. It is preprepared values, not necessarily taking place in a particular situation. Therefore, frame structure contains: name of frame, inheritance pointers (if frame structure is hierarchical) name of slots, values data type and values of slots. In most frame structures value of slot may other frames, thereby realizing various relations between concepts and procedures (Procedural attachments), implementing the procedural component in knowledge representation.

Features and benefits of frame model make it potentially used in our developing IS. Also we can use a frame script to represent the objectives of the organization of user interaction with the web application. Thus, working with IS we are able to refine the relevant attributes of the frame: requirements for a web application, characteristics of the target user and set of recommendations.

The production model is one of the most common tools of knowledge representation in IS with its clarity, modularity (the ease of making changes and additions) and convenience for output [9]. In general, production model (P) can be represented as follows:

$$P =< S; L; A_p \rightarrow B_p; Q, \qquad (2)$$

where S is description of class situations; L is a condition in which products are activated; $A_p \rightarrow B_p$ is the core of the product; Q - postcondition production rule.

Among the disadvantages of production model include:

1. simple conjuncts structure, no relationship between them, which leads to a small use model to describe the complex field of knowledge domain;
2. the complexity of the rules which may to conflict each other and management of priority of their implementation.

To avoid these disadvantages we can use production model in symbiosis means of knowledge representation, which has recently gained popularity in the formalization of knowledge [6]. In this situation production model may be included in declarative component of knowledge representation: frames will set the structure of field of knowledge and rules of production model used to fill the value of frame instances. Also the solver, one of the main components of the IS must be able to work with such hybrid model of knowledge representation, as well as controlling the sequence of execution of rules.

7 Classes related to knowledge in HCI area and UI

Based on the architecture of IS, first class of ontology was HCI engineering task, which has relationship with the classes representing input (Target user and Requirement and output (Guideline (Recommendation) and web interface design) information. In addition, the value of project context slot can be set, which includes any class of ontology (a child of the meta-class THING). The overall structure of the class HCI engineering task is shown in **Fig. 2**.

Law, Principle and Guideline are ontology classes, united in abstract meta class called HCI knowledge. It is known that the success of HCI engineering depends on the deep understanding of interaction context and human behavior patterns, which gives engineer the opportunity to make informed choices of a practical recommendations of several different or its contradictory [14]. This gives particular importance to organize knowledge in a set of recommendations issued by IS, which corresponds to subsystem of explanations and makes it appropriate to provide for:

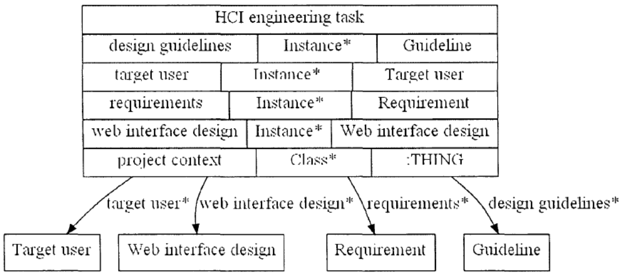

Fig. 2: HCI engineering task class structure

1. Possibility of establishing links between various levels of HCI knowledge and knowledge of one level;
2. possibility of establishing links to frame instance;
3. Possibility of establishing a link between an instance and context;
4. Ability to specify the ”efficiency” or practical significance for recommendations.

Based on points 1. and 2. above, in ontology we added classes Finding, Source and Reference and also corresponding relationship with all classes. Despite this all classes related to HCI knowledge has been combined into abstract class called HCI knowledge representation class. **(Fig. 3)**.

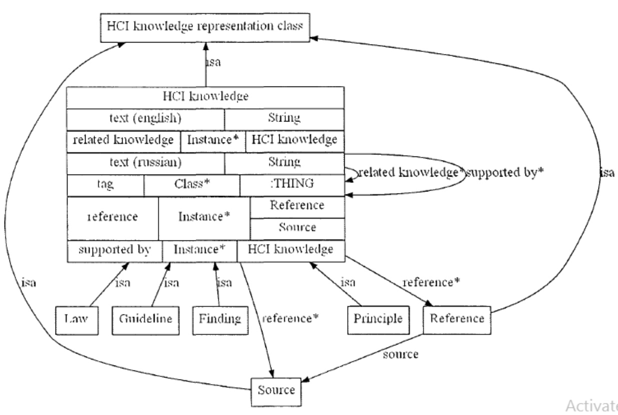

Fig. 3: HCI engineering task class structure

Class *Finding* reflects some fact or empirical evidence obtained in research or practice in HCI area, class *Source* usually corresponds to a scientific publication, i.e. article or book. Based on the points 3. and 4. above there are slots *tag* and *efficiency*. If the first one is the attribute of knowledge at all levels (i.e. class HCI knowledge class), then the second refers only to the Guideline class. Slot tag is used to establish relation between an instance of knowledge and context of projected interaction. The value of this slot can be any classes of ontology. Slot efficiency reflects the evaluation of efficiency of recommendations in HCI engineering;

The end result of activity on designing interaction is interface which created as a result of compromise solutions aimed at meeting the requirements and considering technological and other limitations [15]. Designer can make a decision based on practical recommendations or a higher level of knowledge existing in HCI. But separate rules such decisions are not formalized.

Today web applications can be divided into two components: content and design, which is closely connected with each other. High-quality presentation of content, organization of its user-friendly and adaptation for ”scanning” style of perception is an important factor to improve the usability of web applications [16].

8 Conclusion

We analyzed and built architecture methods and techniques for HCI. The application of intelligent system supports HCI engineering allow us to reduce requirements for qualification of engineers and the error probability of interaction design at initial stages of the process during web application development and also improve the quality of interaction for specific categories of users.

In further development of the topic, we are going to update our knowledge base to represent more precise models of recommendations to HCI engineer. Also we are going to provide more detailed validation of knowledge base of existing web application.

References

[1] Nielsen, J. Jakob Nielsen’s Alertbox, January 22, 2008: Usability ROI Declining, But Still Strong. // access performed on 03.04.2015 from http://www.useit.com/alertbox/roi.html.
[2] Henninger, S., Ashokkumar, P. An Ontology-Based Infrastructure for Usability Design Patterns. // In Proc. Semantic Web Enabled Software Engineering (SWESE), Galway, Ireland, 2005. pp. 41-55.
[3] Internet World Stats. Europe Internet Usage Stats Facebook and Population Statistics. // access performed on 05.04.2015.
[4] Fund ”Public Opinion”. Thirty-third edition of the regular newsletter ”The Internet in Kazakhstan”. Issue 33 Spring 2011.
[5] Nielsen, J. Jakob Nielsen’s Alertbox, November 6, 2006: 100 Million Websites. // access performed on 05.04.2015 from http://www.useit.com/alertbox/web-growth.html.
[6] Netcraft. December 2011 Web Server Survey. // access performed on 05.04.2015 from http://news.netcraft.com/archives/2011 /12/09/december-2011 -web-serversurvey.html.

[7] Nielsen, J. Jakob Nielsen's Alertbox, March 19, 2000: Why You Only Need to Test With 5 Users. // access performed on 05.04.2015 from http://www.useit.com/alertbox/20000319.html.
[8] Brooks F. Collaboration and telecollaboration in design. 7th Turing Lecture, IEEE. Savoy Place, London, 20 Jan 2005.
[9] Glass, R.L. Facts and Fallacies of Software Engineering // Addison Wesley, 2002. pp. 224.
[10] ISO / IEC 9126-93. Information technology. Assessment quality of software products. Quality characteristics and guidelines for their use.
[11] Nielsen, J. Jakob Nielsen's Alertbox, January 21, 2001: Usability Metrics. // access performed on 05.04.2015 from http://www.useit.com/alertbox/20010121.html.
[12] ISO. ISO/IEC 25062:2006 "Common Industry Format (CIF) for usability test reports"
[13] Minsky, . A framework for representing knowledge // In The Psychology of Computer Vision, McGraw-Hill: P. Winston, 1975. pp. 211-277.
[14] Nielsen, J. Jakob Nielsen's Alertbox, March 1, 2004: Risks of Quantitative Studies. // access performed on 05.04.2015 from http://www.useit.com/alertbox/20040301.html.
[15] Howard, S. Trade-off decision making in user interface design. // Behavior and Information Technology, 16 (2), 1997. pp. 98-109.
[16] Nielsen, J. Jakob Nielsen's Alertbox, March 12, 2007: 10 High-Profit Redesign Priorities. // access performed on 05.04.2015 from http://www.useit.com/alertbox/high-roi.html.
[17] ISO. ISO/IEC TR 9126-4:2004 Software engineering and Product quality
[18] Nielsen, J. Usability Engineering. // Morgan Kaufmann, San Francisco, USA, 1994. pp. 362
[19] Vanderdonckt, J. Development milestones towards a tool for working with guidelines.//Interacting with Computers, 12, 1999. pp. 81-118.
[20] Gruber, T.R. A translation approach to portable ontologies. // Knowledge Acquisition, 5 (2), 1993. pp. 199-220.
[21] Mirseidova Sh. , Atymtayeva L. Definition of Software Metrics for Software Project Development by Using Fuzzy Sets and Logic // Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems, 20-24 November 2012, Kobe, Japan, pp. 272-276
[22] Atymtayeva L., Kozhakhmet K., Bortsova G. Building a Knowledge Base for Expert System in Information Security. // Proceedings of the 14th International Symposium on Advanced Intelligent Systems (ISIS2013), 13-16 November 2013, Daejeon, Korea, // Springer Journal Advances in Intelligent Systems and Computing, Volume 270 "Soft Computing in Artificial Intelligence ", pp. 57-77

Madi Sheriyev Madi Sheriyev received the bachelor degree in Information Systems at International Information Technology University in Almaty, Kazakhstan. The Author has been pursuing his research work under the guidance of Lyazzat Atymtayeva, Doctor of Science degree in Mechanics, Mathematics and Computer Science. Currently he is studying as master student at KBTU. His research Interest includes Human Computer Interaction, Artificial Intelligence and Semantic Web.

Lyazzat Atymtayeva received the PhD and Doctor of Science degree in Mechanics, Mathematics and Computer Science at al-Farabi Kazakh National University, Kazakhstan. Her research interests are in the areas of mechanics, applied mathematics and computer science including the numerical and rigorous mathematical methods and models for mechanical engineering and computer science, intelligent and expert systems in Information Security, Project Management and HCI. She has published research papers in reputed international journals of mathematical and computer sciences. She is reviewer and editor of international journals in mathematics and information sciences.